Semi-Supervised Learning

Xiaojin Zhu, University of Wisconsin-Madison

Synonyms: Learning from labeled and unlabeled data, transductive learning

Definition

Semi-supervised learning uses both labeled and unlabeled data to perform an otherwise supervised learning or unsupervised learning task.

In the former case, there is a distinction between inductive semi-supervised learning and transductive learning. In inductive semi-supervised learning, the learner has both labeled training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l} \stackrel{iid}{\sim} p(\mathbf{x}, y)$ and unlabeled training data $\{\mathbf{x}_i\}_{i=l+1}^{l+u} \stackrel{iid}{\sim} p(\mathbf{x})$, and learns a predictor $f : \mathcal{X} \mapsto \mathcal{Y}$, $f \in \mathcal{F}$ where $\mathcal{F}$ is the hypothesis space. Here $\mathbf{x} \in \mathcal{X}$ is an input instance, $y \in \mathcal{Y}$ its target label (discrete for classification or continuous for regression), $p(\mathbf{x}, y)$ the unknown joint distribution and $p(\mathbf{x})$ its marginal, and typically $l \ll u$. The goal is to learn a predictor that predicts future test data better than the predictor learned from the labeled training data alone. In transductive learning, the setting is the same except that one is solely interested in the predictions on the unlabeled training data $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$, without any intention to generalize to future test data.

In the latter case, an unsupervised learning task is enhanced by labeled data. For example, in semi-supervised clustering (a.k.a. constrained clustering) one may have a few must-links (two instances must be in the same cluster) and cannot-links (two instances cannot be in the same cluster) in addition to the unlabeled instances to be clustered; in semi-supervised dimensionality reduction one might have the target low-dimensional coordinates on a few instances.

This entry will focus on the former case of learning a predictor.

Motivation and Background

Semi-supervised learning is initially motivated by its practical value in learning faster, better, and cheaper. In many real world applications, it is relatively easy to acquire a large amount of unlabeled data $\{\mathbf{x}\}$. For example, documents can be crawled from the Web, images can be obtained from surveillance cameras, and speech can be collected from broadcast. However, their corresponding labels $\{y\}$ for the prediction task, such as sentiment orientation, intrusion detection, and phonetic transcript, often requires slow human annotation and expensive laboratory experiments. This labeling bottleneck results in a scarce of labeled data and a surplus of unlabeled data. Therefore, being able to utilize the surplus unlabeled data is desirable.

Recently, semi-supervised learning also finds applications in cognitive psychology as a computational model for human learning. In human categorization and concept forming, the environment provides unsupervised data (e.g., a child watching surrounding objects by herself) in addition to labeled data from a teacher (e.g., Dad points to an object and says "bird!"). There is evidence that human beings can combine labeled and unlabeled data to facilitate learning.

The history of semi-supervised learning goes back to at least the 70s, when self-training, transduction, and Gaussian mixtures with the EM algorithm first emerged. It enjoyed an explosion of interest since the 90s, with the development of new algorithms like co-training and transductive support vector machines, new applications in natural language processing and computer vision, and new theoretical analyses. More discussions can be found in section 1.1.3 in [7].

Theory

It is obvious that unlabeled data $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$ by itself does not carry any information on the mapping $\mathcal{X} \mapsto \mathcal{Y}$. How can it help us learn a better predictor $f : \mathcal{X} \mapsto \mathcal{Y}$? Balcan and Blum pointed out in [2] that the key lies in an implicit ordering of $f \in \mathcal{F}$ induced by the unlabeled data. Informally, if the implicit ordering happens to rank the target predictor f^* near the top, then one needs less labeled data to learn f^*. This idea will be formalized later on using PAC learning bounds. In other contexts, the implicit ordering is interpreted as a prior over $\mathcal{F}$ or as a regularizer.

A semi-supervised learning method must address two questions: what implicit ordering is induced by the unlabeled data, and how to algorith-

mically find a predictor near the top of this implicit ordering and fits the labeled data well. Many semi-supervised learning methods have been proposed, with different answers to these two questions [15, 7, 1, 10]. It is impossible to enumerate all methods in this entry. Instead, we present a few representative methods.

Generative Models

This semi-supervised learning method assumes the form of joint probability $p(\mathbf{x}, y \mid \theta) = p(y \mid \theta)p(\mathbf{x} \mid y, \theta)$. For example, the class prior distribution $p(y \mid \theta)$ can be a multinomial over $\mathcal{Y}$, while the class conditional distribution $p(\mathbf{x} \mid y, \theta)$ can be a multivariate Gaussian in $\mathcal{X}$ [6, 9]. We use $\theta \in \Theta$ to denote the parameters of the joint probability. Each θ corresponds to a predictor f_θ via Bayes rule:

$$f_\theta(\mathbf{x}) \equiv \operatorname*{argmax}_y p(y \mid \mathbf{x}, \theta) = \operatorname*{argmax}_y \frac{p(\mathbf{x}, y \mid \theta)}{\sum_{y'} p(\mathbf{x}, y' \mid \theta)}.$$

Therefore, $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$.

What is the implicit ordering of f_θ induced by unlabeled training data $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$? It is the large to small ordering of log likelihood of θ on unlabeled data:

$$\log p(\{\mathbf{x}_i\}_{i=l+1}^{l+u} \mid \theta) = \sum_{i=l+1}^{l+u} \log \left(\sum_{y \in \mathcal{Y}} p(\mathbf{x}_i, y \mid \theta) \right)$$

The top ranked f_θ is the one whose θ (or rather the generative model with parameters θ) best fits the unlabeled data. Therefore, this method assumes that the form of the joint probability is correct for the task.

To identify the f_θ that both fits the labeled data well and ranks high, one maximizes the log likelihood of θ on both labeled and unlabeled data:

$$\operatorname*{argmax}_\theta \log p(\{\mathbf{x}_i, y_i\}_{i=1}^{l} \mid \theta) + \lambda \log p(\{\mathbf{x}_i\}_{i=l+1}^{l+u} \mid \theta),$$

where λ is a balancing weight. This is a non-concave problem. A local maximum can be found with the Expectation-Maximization (EM) algorithm, or other numerical optimization methods.

Semi-Supervised Support Vector Machines

This semi-supervised learning method assumes that the decision boundary $f(\mathbf{x}) = 0$ is situated in a low-density region (in terms of unlabeled data)

between the two classes $y \in \{-1, 1\}$ [12, 8]. Consider the following hat loss function on an unlabeled instance $\mathbf{x}$:

$$\max(1 - |f(\mathbf{x})|, 0)$$

which is positive when $-1 < f(\mathbf{x}) < 1$, and zero outside. The hat loss thus measures the violation in (unlabeled) large margin separation between f and $\mathbf{x}$. Averaging over all unlabeled training instances, it induces an implicit ordering from small to large over $f \in \mathcal{F}$:

$$\frac{1}{u} \sum_{i=l+1}^{l+u} \max(1 - |f(\mathbf{x})|, 0).$$

The top ranked f is one whose decision boundary avoids most unlabeled instances by a large margin.

To find the f that both fits the labeled data well and ranks high, one typically minimizes the following objective:

$$\operatorname*{argmin}_f \frac{1}{l} \sum_{i=1}^{l} \max(1 - y_i f(\mathbf{x}_i), 0) + \lambda_1 \|f\|^2 + \lambda_2 \frac{1}{u} \sum_{i=l+1}^{l+u} \max(1 - |f(\mathbf{x})|, 0),$$

which is a combination of the objective for supervised support vector machines, and the average hat loss. Algorithmically, the optimization problem is difficult because the hat loss is non-convex. Existing solutions include semi-definite programming relaxation, deterministic annealing, continuation method, concave-convex procedure (CCCP), stochastic gradient descent, and Branch and Bound.

Graph-Based Models

This semi-supervised learning method assumes that there is a graph $G = \{V, E\}$ such that the vertices V are the labeled and unlabeled training instances, and the undirected edges E connect instances i, j with weight w_{ij} [4, 14, 3]. The graph is sometimes assumed to be a random instantiation of an underlying manifold structure that supports $p(\mathbf{x})$. Typically, w_{ij} reflects the proximity of $\mathbf{x}_i, \mathbf{x}_j$. For example, the Gaussian edge weight function defines $w_{ij} = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2\right)$. As another example, the kNN edge weight function defines $w_{ij} = 1$ if $\mathbf{x}_i$ is within the k nearest neighbors of $\mathbf{x}_j$ or vice versa, and $w_{ij} = 0$ otherwise. Other commonly used edge weight functions include ϵ-radius neighbors, b-matching, and combinations of the above.

Large w_{ij} implies a preference for the predictions $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ to be the same. This can be formalized by the graph energy of a function f:

$$\sum_{i,j=1}^{l+u} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2.$$

The graph energy induces an implicit ordering of $f \in \mathcal{F}$ from small to large. The top ranked function is the smoothest with respect to the graph (in fact, it is any constant function). The graph energy can be equivalently expressed using the so-called unnormalized graph Laplacian matrix. Variants including the normalized Laplacian and the powers of these matrices.

To find the f that both fits the labeled data well and ranks high (i.e., being smooth on the graph or manifold), one typically minimizes the following objective:

$$\operatorname*{argmin}_f \frac{1}{l}\sum_{i=1}^{l} c(f(\mathbf{x}_i), y_i) + \lambda_1\|f\|^2 + \lambda_2 \sum_{i,j=1}^{l+u} w_{ij}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2,$$

where $c(f(\mathbf{x}), y)$ is a convex loss function such as the hinge loss or the squared loss. This is a convex optimization problem with efficient solvers.

Co-training and Multiview Models

This semi-supervised learning method assumes that there are multiple, different learners trained on the same labeled data, and these learners agree on the unlabeled data. A classic algorithm is co-training [5]. Take the example of web page classification, where each web page $\mathbf{x}$ is represented by two subsets of features, or "views" $\mathbf{x} = \langle \mathbf{x}^{(1)}, \mathbf{x}^{(2)} \rangle$. For instance, $\mathbf{x}^{(1)}$ can represent the words on the page itself, and $\mathbf{x}^{(2)}$ the words on the hyperlinks (on other web pages) pointing to this page. The co-training algorithm trains two predictors: $f^{(1)}$ on $\mathbf{x}^{(1)}$ (ignoring the $\mathbf{x}^{(2)}$ portion of the feature) and $f^{(2)}$ on $\mathbf{x}^{(2)}$, both initially from the labeled data. If $f^{(1)}$ confidently predicts the label of an unlabeled instance $\mathbf{x}$, then the instance-label pair $(\mathbf{x}, f^{(1)}(\mathbf{x}))$ is added to $f^{(2)}$'s labeled training data, and vice versa. Note this promotes $f^{(1)}$ and $f^{(2)}$ to predict the same on $\mathbf{x}$. This repeats so that each view teaches the other. Multiview models generalize co-training by utilizing more than two predictors, and relaxing the requirement of having separate views [11]. In either case, the final prediction is obtained from a (confidence weighted) average or vote among the predictors.

To define the implicit ordering on the hypothesis space, we need a slight extension. In general, let there be m predictors $f^{(1)}, \ldots f^{(m)}$. Now let a

hypothesis be an m-tuple of predictors $\langle f^{(1)}, \dots f^{(m)} \rangle$. The disagreement of a tuple on the unlabeled data can be defined as

$$\sum_{i=l+1}^{l+u} \sum_{u,v=1}^{m} c(f^{(u)}(\mathbf{x}_i), f^{(v)}(\mathbf{x}_i)),$$

where $c()$ is a loss function. Typical choices of $c()$ are the 0-1 loss for classification, and the squared loss for regression. Then the disagreement induces an implicit ordering on tuples from small to large.

It is important for these m predictors to be of diverse types, and have different inductive biases. In general, each predictor $f^{(u)}, u = 1 \dots m$ may be evaluated by its individual loss function $c^{(u)}$ and regularizer $\Omega^{(u)}$. To find a hypothesis (i.e,. m predictors) that fits the labeled data well and ranks high, one can minimize the following objective:

$$\begin{aligned} \underset{\langle f^{(1)}, \dots f^{(m)} \rangle}{\operatorname{argmin}} \quad & \sum_{u=1}^{m} \left(\frac{1}{l} \sum_{i=1}^{l} c^{(u)}(f^{(u)}(\mathbf{x}_i), y_i) + \lambda_1 \Omega^{(u)}(f^{(u)}) \right) \\ & + \lambda_2 \sum_{i=l+1}^{l+u} \sum_{u,v=1}^{m} c(f^{(u)}(\mathbf{x}_i), f^{(v)}(\mathbf{x}_i)). \end{aligned}$$

Multiview learning typically optimizes this objective directly. When the loss functions and regularizers are convex, numerical solution is relatively easy to obtain. In the special cases when the loss functions are the squared loss, and the regularizers are squared ℓ_2 norms, there is a closed form solution. On the other hand, the co-training algorithm, as presented earlier, optimizes the objective indirectly with the iterative procedure. One advantage of co-training is that the algorithm is a wrapper method, in that it can use any "blackbox" learners $f^{(1)}$ and $f^{(2)}$ without the need to modify the learners.

A PAC Bound for Semi-Supervised Learning

Previously, we presented several semi-supervised learning methods, each induces an implicit ordering on the hypothesis space using the unlabeled training data, and each attempts to find a hypothesis that fit the labeled training data well as well as rank high in that implicit ordering. We now present a theoretical justification on why this is a good idea. In particular, we present a uniform convergence bound by Balcan and Blum (Theorem 11 in [2]). Alternative theoretical analyses on semi-supervised learning can be found by following the recommended reading.

First, we introduce some notations. Consider the 0-1 loss for classification. Let $c^* : \mathcal{X} \mapsto \{0, 1\}$ be the unknown target function, which may not be in $\mathcal{F}$. Let $err(f) = \mathbb{E}_{\mathbf{x}\sim p}[f(\mathbf{x}) \neq c^*(\mathbf{x})]$ be the true error rate of a hypothesis f, and $\widehat{err}(f) = \frac{1}{l}\sum_{i=1}^{l} f(\mathbf{x}_i) \neq c^*(\mathbf{x}_i)$ be the empirical error rate of f on the labeled training sample. To characterize the implicit ordering, we defined an "unlabeled error rate" $err_{unl}(f) = 1 - \mathbb{E}_{\mathbf{x}\sim p}[\chi(f, \mathbf{x})]$, where the *compatibility function* $\chi : \mathcal{F} \times \mathcal{X} \mapsto [0, 1]$ measures how "compatible" f is to an unlabeled instance $\mathbf{x}$. As an example, in semi-supervised support vector machines, if $\mathbf{x}$ is far away from the decision boundary produced by f, then $\chi(f, \mathbf{x})$ is large; but if $\mathbf{x}$ is close to the decision boundary, $\chi(f, \mathbf{x})$ is small. In this example, a large $err_{unl}(f)$ then means that the decision boundary of f cuts through dense unlabeled data regions, and thus f is undesirable for semi-supervised learning. In contrast, a small $err_{unl}(f)$ means that the decision boundary of f lies in a low density gap, which is more desirable. In theory, the implicit ordering on $f \in \mathcal{F}$ is to sort $err_{unl}(f)$ from small to large. In practice, we use the empirical unlabeled error rate $\widehat{err}_{unl}(f) = 1 - \frac{1}{u}\sum_{i=l+1}^{l+u} \chi(f, \mathbf{x}_i)$.

Our goal is to show that if an $f \in \mathcal{F}$ "fits the labeled data well and ranks high", then f is almost as good as the best hypothesis in $\mathcal{F}$. Let $t \in [0, 1]$. We first consider the best hypothesis f_t^* in the subset of $\mathcal{F}$ that consists of hypotheses whose unlabeled error rate is no worse than t: $f_t^* = \text{argmin}_{f'\in\mathcal{F}, err_{unl}(f')\leq t}\, err(f')$. Obviously, $t = 1$ gives the best hypothesis in the whole $\mathcal{F}$. However, the nature of the guarantee has the form $err(f) \leq err(f_t^*) + \text{EstimationError}(t) + c$, where the EstimationError term increases with t. Thus, with $t = 1$ the bound can be loose. On the other hand, if t is close to 0, EstimationError(t) is small, but $err(f_t^*)$ can be much worse than $err(f_{t=1}^*)$. The bound will account for the optimal t.

We introduce a few more definitions. Let $\mathcal{F}(f) = \{f' \in \mathcal{F} : \widehat{err}_{unl}(f') \leq \widehat{err}_{unl}(f)\}$ be the subset of $\mathcal{F}$ with empirical error no worse than that of f. As a complexity measure, let $[\mathcal{F}(f)]$ be the number of different partitions of the first l unlabeled instances $\mathbf{x}_{l+1} \ldots \mathbf{x}_{2l}$, using $f \in \mathcal{F}(f)$. Finally, let $\hat{\epsilon}(f) = \sqrt{\frac{24}{l}\log(8[\mathcal{F}(f)])}$. Then we have the following agnostic bound (meaning that c^* may not be in $\mathcal{F}$, and $\widehat{err}_{unl}(f)$ may not be zero for any $f \in \mathcal{F}$):

Theorem 1. *Given l labeled instances and sufficient unlabeled instances, with probability at least $1 - \delta$, the function*

$$f = \underset{f'\in\mathcal{F}}{\text{argmin}}\, \widehat{err}(f') + \hat{\epsilon}(f')$$

satisfies the guarantee that

$$err(f) \leq \min_t(err(f_t^*) + \hat{\epsilon}(f_t^*)) + 5\sqrt{\frac{log(8/\delta)}{l}}.$$

If a function f fits the labeled data well, it has a small $\widehat{err}(f)$. If it ranks high, then $\mathcal{F}(f)$ will be a small set, consequently $\hat{\epsilon}(f)$ is small. The argmin operator identifies the best such function during training. The bound account for the minimum of all possible t tradeoffs. Therefore, we see that the "lucky" case is when the implicit ordering is good such that $f_{t=1}^*$, the best hypothesis in $\mathcal{F}$, is near the top of the ranking. This is when semi-supervised learning is expected to perform well. Balcan and Blum also give results addressing the key issue of how much *unlabeled* data is needed for $\widehat{err}_{unl}(f)$ and $err_{unl}(f)$ to be close for all $f \in \mathcal{F}$.

Applications

Because the type of semi-supervised learning discussed in this entry has the same goal of creating a predictor as supervised learning, it is applicable to essentially any problems where supervised learning can be applied. For example, semi-supervised learning has been applied to natural language processing (word sense disambiguation [13], document categorization, named entity classification, sentiment analysis, machine translation), computer vision (object recognition, image segmentation), bioinformatics (protein function prediction), and cognitive psychology. Follow the recommended reading for individual papers.

Future Directions

There are several directions to further enhance the value semi-supervised learning. First, we need guarantees that it will outperform supervised learning. Currently, the practitioner has to manually choose a particular semi-supervised learning method, and often manually set learning parameters. Sometimes, a bad choice that does not match the task (e.g., modeling each class with a Gaussian when the data does not have this distribution) can make semi-supervised learning worse than supervised learning. Second, we need methods that benefit from unlabeled when l, the size of labeled data, is large. It has been widely observed that the gain over supervised learning is the largest when l is small, but diminishes as l increases. Third, we need good ways to combine semi-supervised learning and active learning. In

natural learning systems such as humans, we routinely observe unlabeled input, which often naturally leads to questions. And finally, we need methods that can efficiently process massive unlabeled data, especially in an online learning setting.

Cross References

active learning, classification, constrained clustering, dimensionality reduction, online learning, regression, supervised learning, unsupervised learning

Recommended Reading

[1] S. Abney. *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC, 2007.

[2] M.-F. Balcan and A. Blum. A discriminative model for semi-supervised learning. *Journal of the ACM*, 2009.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, November 2006.

[4] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proc. 18th International Conf. on Machine Learning*, 2001.

[5] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1998.

[6] V. Castelli and T. Cover. The exponential value of labeled samples. *Pattern Recognition Letters*, 16(1):105–111, 1995.

[7] O. Chapelle, A. Zien, and B. Schölkopf, editors. *Semi-supervised learning*. MIT Press, 2006.

[8] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. 16th International Conf. on Machine Learning*, pages 200–209. Morgan Kaufmann, San Francisco, CA, 1999.

[9] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[10] M. Seeger. Learning with labeled and unlabeled data. Technical report, University of Edinburgh, 2001.

[11] V. Sindhwani, P. Niyogi, and M. Belkin. A co-regularized approach to semi-supervised learning with multiple views. In *Proc. of the 22nd ICML Workshop on Learning with Multiple Views*, August 2005.

[12] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[13] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[14] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *The 20th International Conference on Machine Learning (ICML)*, 2003.

[15] X. Zhu and A. B. Goldberg. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.